1. Computer Assisted Training

Inform offers the Explorer Edition to provide designers and engineers with a low-cost, practical start in fuzzy technologies. The *fuzzy*TECH Explorer Edition is a MS Windows™-based software package that comes with animated simulations of real-world control applications for interactive fuzzy system training. No software other than MS-Windows™ is required to run the Explorer.

2. Design Courses

To get system designers started with the development, optimization and implementation of fuzzy logic systems, Inform provides seminars and hands-on workshops throughout the world. Already more than 2,500 designers and managers have attended Inform's one-day fuzzy logic seminars, two-day workshops and five-day design courses. Contact Inform for dates and locations.

3. In-House Workshops

To speed up the introduction of fuzzy technologies in design teams, Inform offers specialized introductory in-house workshops for customers. These workshops may be customized by Inform to the specific interest areas of the customer. Contact Inform for a price quotation.

4. Accelerated Technology Access Program (ATAP)

Inform's third level of support is the Accelerated Technology Access Program (ATAP). In this support program, an Inform fuzzy logic specialist joins the client's design team to provide on-the-job training for a limited period of time, usually three weeks. This cooperative effort produces a working prototype for a section of the client's application. The ATAP has two advantages: the client's designers become practiced in the design methodologies of fuzzy logic and the resulting demonstrator proves the feasibility of fuzzy logic for the client's application. The ATAP is offered worldwide by Inform and its partners.

5. Extended Development Support Program (EDSP)

To extend the results of an ATAP into a full-blown application, Inform provides the Extended Development Support Program (EDSP) at customers request. In contrast to the ATAP, the duration of an EDSP varies with the type of development project.

*fuzzy*TECH™ Questions & Answers

INFORM SOFTWARE CORPORATION

What Is Fuzzy Logic?

- A technology which enhances model-based system designs using both intuition and engineering heuristics. Fuzzy logic uses elements of everyday language to represent desired system behaviour, thus circumventing the need for rigorous mathematical modelling.
- An efficient way of designing, optimizing and maintaining highly complex systems transparently.
- A new paradigm of systems engineering which helps achieve robust and fault-friendly systems.
- An enabling technology

Fuzzy Logic – What It NOT Is?

- The solution to ALL problems. Some problems are better solved with conventional methods.
- A "fuzzy" method. Fuzzy technology is based on fuzzy set theory – a recognized and well-studied mathematical discipline.
- A methodology that can be acquired in a minute.

When Should Fuzzy Logic be Used?

- If no adequate mathematical model for a given problem is easily found.
- If non-linearities, time-constraints or multiple parameters exist.
- If engineering know-how about the given problem is available or can be acquired during the design process.

When Should Fuzzy Logic NOT be Used?

- When the problem can be easily solved using conventional control techniques, such as a PID controller.
- If there is a simple, clear-defined and easy-to-solve mathematical model for the given problem.
- When the problem can not be solved at all. There are some problems with which even fuzzy logic can not help you.

What are the Benefits of using Fuzzy Logic?

- System performance is improved by supplementing conventional modeling with engineering expertise.
- Fuzzy Logic enables new functionality to existing designs
- Using advanced development tools, it speeds up the development cycle and time to market.

Why Tools for Fuzzy Logic?

- Coding fuzzy logic on standard microcontrollers using conventional programming languages may be a good learning experience, but doing this for system design can be tiresome and time-consuming. Fuzzy-logic design tools implement the fuzzy algorithms on just about any standard hardware.
- *fuzzy*TECH tools offer online/offline simulation and optimization features to speed up the development cycle.
- *fuzzy*TECH tools allow a full graphical development of the system. This expedites design and results in a structure that is easy to comprehend and maintain.

What Is *fuzzy*TECH?

- A full-graphical CASE tool that supports all design steps for fuzzy system engineering.
- A simulation and optimization tool for fuzzy systems. It is easily connected with conventional design and simulation tools. It displays system performance in multiple ways to give you efficient optimization options.
- A code generator. To support most existing hardware, the designed system may be generated as C-code. To achieve the most optimized code, assembly code generation for industry microcontroller families are provided.

What are the Design Steps using *fuzzy*TECH?

1. System definition: define linguistic variables describing input/output values and IF-THEN rules representing the control strategy.
2. Offline optimization: analyze system performance interactively by using either sample data or the model simulation.
3. Online optimization: connect the system to the process hardware and optimize the controller during runtime.
4. Implementation: select the appropriate code generator for your target hardware and commit to firmware.

What Is the Programming Language of *fuzzy*TECH?

- Internally, *fuzzy*TECH is based on FTL, the Fuzzy Technology Language. FTL is a hardware-independent object-oriented description language for fuzzy system engineering.
- FTL is an open standard supported by various hardware vendors and may be extended to incorporate individual user extensions.

What are the Different Editions of *fuzzy*TECH?

- The Explorer Edition is the ideal training kit. It provides you with an instant practical introduction to fuzzy-logic engineering.
- The various MCU Editions generate optimized code for standard microcontrollers. Their code efficiency allows fuzzy techniques to be integrated into most designs.
- The *fuzzy*TECH Online Edition allows you to cross-debug a running system and modify it on-the-fly. In real-time control applications, this greatly accelerates the optimization phase of system design.

What Is the Origin of *fuzzy*TECH?

- Prof. Hans Zimmermann, who for over 20 years has been one of the most recognized researchers in the field of fuzzy set theory, began developing the methods and tools for implementing advanced fuzzy technologies.
- The software tool is named *fuzzy*TECH. Produced by INFORM GmbH in Germany, it is the most successful fuzzy system engineering tool in Europe.
- In October 1992, Inform Software Corp. of Evanston, IL, introduced the new *fuzzy*TECH Release 3.0 to U.S. customers.

*fuzzy*BUILDER Kits

Your Total H/W & S/W "Turn Key" Project Design Kit With 16 Bit Embedded Control Performance!

(Product order codes and pricing on the reverse side)

There's nothing as fast and easy ! Here's why:

***fuzzy*TECH S/W is a CASE Tool:** You develop your system in a total graphical development environment. It allows the engineer to work at a higher level of abstraction, *designing your system from the top down* -- NOT from the bottom up! *Spend your time solving the problem, ... NOT programming the system! fuzzy*TECH tools generate highly optimized **16 Bit** 'C' code with ASM calls to the fuzzy kernel. Performance is over 3X faster than straight 'C' code!

Click your mouse on "Compile" and the entire system code is generated!

***Ap*BUILDER writes the peripheral initialization code for the Embedded Controller.** With windowed screens it guides you through each peripheral and prevents programming mistrakes. It generates either C or ASM code written as a 'function'. So, when the fuzzy logic system needs the resources of the microcontroller, its just a C call to the code functions generated by *Ap*BUILDER!

Click your mouse to generate the microcontroller peripheral initialization code with *Ap*BUILDER. Now the fuzzy system is interfaced to the chip!

The 196KD Target Board interfaces the system to your project hardware. Download your system at 57.6 Kbaud with the windowed Symbolic DeBug Monitor. Use the kit's Demo ASM or iC-96 compiler (Demo iC-96 compiler available with product registration) to modify code and write additional S/W for your application. Interface the Target Board to your project hardware via the edge connector pins. Program your system code into FLASH or EPROM and you are ready to run a stand alone system.

Interface the 196KD Target Board to external H/W, pull one jumper pin and... the board becomes a stand alone system!

Now That's Working Smarter ... NOT Harder !!!

Product Ordering Information

Product	Order Code	Price
Project Builder 196 Kit	**PROJBLD196KBCD**	$ 196.00

fuzzy*BUILDER 196 Kits:** ***(total Hardware and Software Kits for the 196KB/KC/KD)
(each fuzzyBUILDER Kit Includes the Project Builder 196 Kit)

Project Builder 196 & *fuzzy*TECH 3.0 MCU-96 Explorer	**fuzzyBLDREXP96**	$ 396.00
Project Builder 196 & *fuzzy*TECH 3.0 MCU-96 Edition	**fuzzyBLDRMCU96**	$ 2096.00
Project Builder 196, *fuzzy*TECH 3.0 MCU-96 Edition & NeuroFuzzy Module	**fuzzyBLDRNFK96** (Available after 12/93)	$ 2996.00
Project Builder 196, *fuzzy*TECH 3.0 MCU-96 Edition & Real Time Remote Cross Debugger	**fuzzyBLDRXDB96** (Available after 12/93)	$ 4896.00

Fuzzy Logic Software

*fuzzy*TECH 3.0 Explorer-96 Edition	**fuzzyTECHEXP96**	$ 215.60
*fuzzy*TECH 3.0 MCU-96 Edition	**fuzzyTECHMCU96**	$ 2056.00
Real Time Remote Cross Debugger	**fuzzyRTXDBUG96**	$ 2936.00 (*)
*fuzzy*TECH 3.0 NeuroFuzzy Module	**NeuralFuzzyMOD**	$ 900.00 (*)
*fuzzy*TECH MCU-51 Edition	**fuzzyTECHMCU51**	$ 1931.51

(*) Denotes Available after 12/93 *(All prices are Intel suggested resale and subject to change)*

For the above products please contact an Intel Distributor. For a list of Intel Distributors call **(800) 628-2283** in the U.S.A. and Canada, or **(916) 356-3105** and request *Fax*BACK™ document # **2589.**

Fuzzy Logic Performance

	7 Rules 2 in / 1 out	20 Rules 2 in / 1 out	20 FAM Rules 2 in / 1 out	80 FAM Rules 3 in / 1 out	500 FAM Rules 8 in / 4 out
80C196	0.19ms/0.22ms	0.28ms/0.33ms	0.29ms/0.34ms	0.43ms/0.50ms	
20MHz	0.73 KB ROM	0.82 KB ROM	0.87 KB ROM	1.27 KB ROM	
	54 Byte RAM	63 Byte RAM	63 Byte RAM	69 Byte RAM	
80C51	1.0ms / 1.2ms	1.4ms / 1.9ms	1.5ms / 2.0ms	4.4ms / 4.9ms	16.7ms / 18.0ms
12MHz	0.45 KB ROM	0.54 KB ROM	0.58 KB ROM	1.0 KB ROM	3.0 KB ROM
	21 Byte RAM*	25 Bytes RAM*	25 Byte RAM*	29 Byte RAM*	103 Byte RAM*

All computing times: Average Case / Worst Case *Register RAM not included
FAM Rules are individually weighted rules. *fuzzy*TECH is a trademark of Inform Software.

Preliminary Copy

*fuzzy*TECH™ 3.0 NeuroFuzzy Module

INFORM SOFTWARE CORPORATION

The *fuzzy*TECH 3.0 NeuroFuzzy Module is an extension module for all *fuzzy*TECH 3.0 Editions. The *fuzzy*TECH 3.0 NeuroFuzzy Module automates the generation and optimization of fuzzy logic systems based on training data input. In most applications system behavior is not only expressed by linguistic knowledge but also represented by sample data. These data sets, consisting of inputs and desired output, represent various states of the process. Typical applications are found in the areas of decision support problems, diagnostics, pattern recognition, data analysis and continuos control applications. Using automated learning methods (e.g. neural learning) the data sets can now be used to both generate and optimize a fuzzy system.

Best of Both Worlds: Learning Fuzzy Logic Systems

The *fuzzy*TECH 3.0 NeuroFuzzy Module combines the advantages of fuzzy systems -- transparent representation of knowledge and the ability to cope with uncertainties -- with the advantages of neural nets -- the ability to learn.
The learning techniques employ neural network learning methods. Up to now, one of the problems of these methods, especially for engineering applications, was that there was no direct access and understanding of the distributed information stored in the network. Therefore, these networks could hardly be optimized at a later time with the addition of new data sets. The solution to this problem is using neural learning methods combined with the concepts of fuzzy logic. Components of a fuzzy logic system can be mapped to different layers of a neural net and vice versa. To accomplish this, the technologies of approximate reasoning used in *fuzzy*TECH are mandatory. Thus the rules as well as membership functions can be modified by the neural training methods, enabling learning as well as the visualization of both the resulting system and the learning procedure.

Include your own User Defined Learning Methods

By taking advantage of an interface to the data structure of *fuzzy*TECH 3.0 the user can implement his own learning or adaptation algorithm using neural net, heuristic, statistic, genetic or other methods. Within the user's algorithms you are free to create and delete rules, adjusting the degrees of support (weight) as well as modifying the configuration of the membership functions through the interface to the system data structure. This gives the user total freedom for custom algorithm development. After linking these algorithms (as DLL's) to the *fuzzy*TECH 3.0 NeuroFuzzy Module they are available with the click of a button.

"Selective" Learning

The fuzzyTECH 3.0 NeuroFuzzy Module is so flexible, it enables you to individually select the parts of the fuzzy logic system where learning should take place. This allows you to preprogram the system with domain knowledge and lock parts of the system where it should be unchangeable, while allowing the reminder of the system to be learned through a learning algorithm. This preprogramming method shortens the learning time by magnitudes and enhances the desired level of control over the results.

*fuzzy*TECH 3.0 NeuroFuzzy Module Specifications

- Add-on Module for all *fuzzy*TECH 3.0 Editions.
- Extension to the internal *fuzzy*TECH debugger enabling the use of all graphical *fuzzy*TECH editors.

Training Modes

- Learning membership function configurations by modifying the definition points of membership functions.
- Learning of selected rules by modifying the degree of support
- Optimize rules and membership functions at the same time.

Learning

• System influence
- Partial optimization of fuzzy systems by specifying objects to be learned.
- Learning of fuzzy logic rules and their degrees of support.
- Learning of membership function configurations.

• Learning Methods
- Learning with modified error back propagation.

• Learning Parameters
- Random, exponential decreasing or constant learning rates.
- Global and local error thresholds.
- Sequential or random sample data processing.

Example data

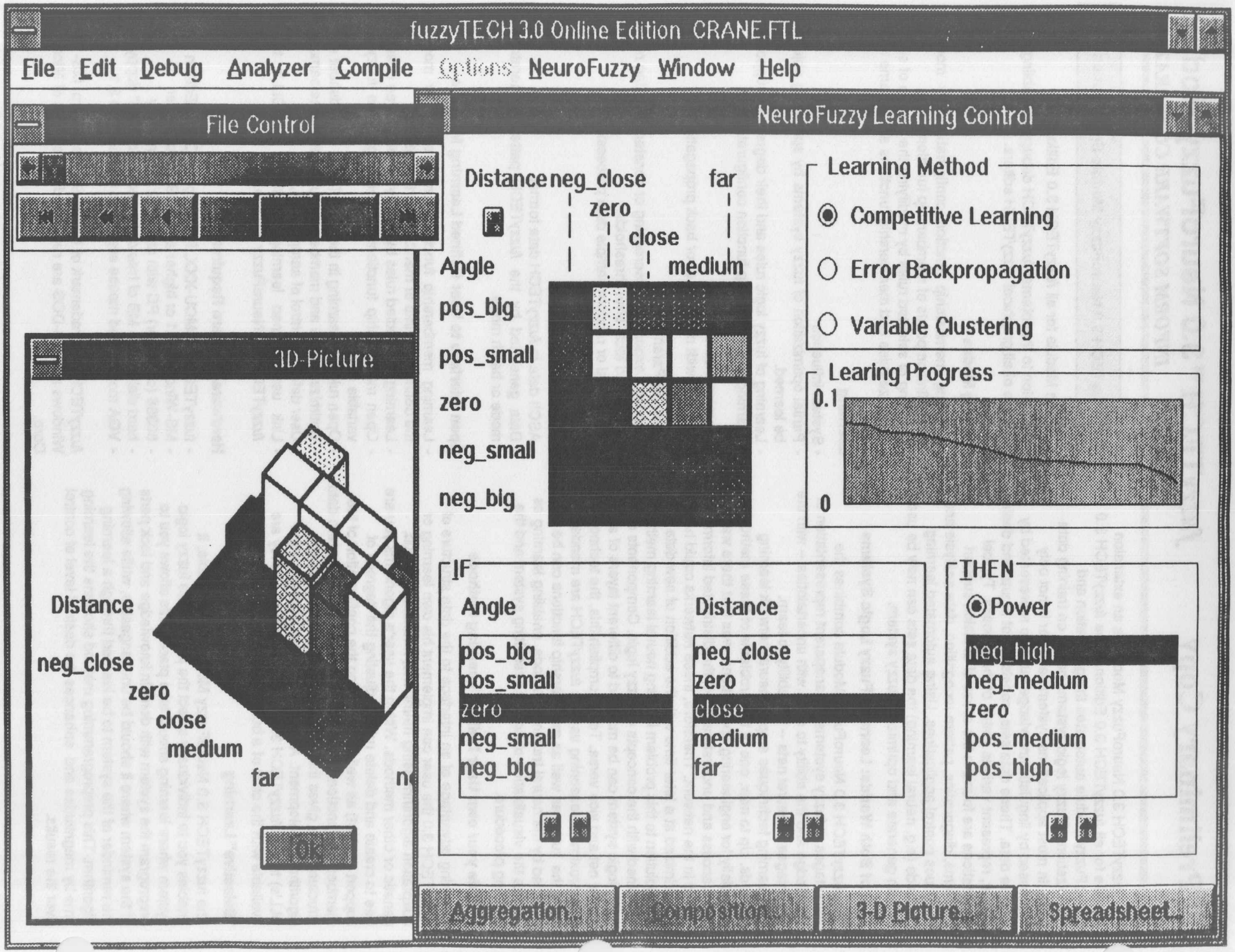
- ASCII data in *fuzzy*TECH data format
- Data generated by the *fuzzy*TECH pattern generator, file mode or batch mode

Open Interface to User Defined Learning Methods

- Learning membership function configurations by modifying the definition points of membership functions.
- Learning of selected rules by modifying the degree of support
- Open membership functions for learning in the *fuzzy*TECH variable editor
- Open rules for learning in the fuzzyTECH spreadsheet editor.
- Optimizing rules and membership functions at the same time.
- User defined control of sample data.
- Link user defined learning algorithms as DLL's to the *fuzzy*TECH 3.0 NeuroFuzzy Module.

Hardware/Software Requirements

- fuzzyTECH 3.0 MCU-XXX, Precompiler, Online Edition
- MS-Windows 3.1 or higher and DOS 5.0 or higher
- 80386 (or higher) PC with at least 2 MB memory
- hard disk with 5 MB of free disk space and a 3,5" floppy
- VGA monitor and mouse supported by Windows 3.1


fuzzyTECH 3.0 Online Edition CRANE.FTL
File Edit Debug Analyzer Compile Options NeuroFuzzy Window Help
File Control
NeuroFuzzy Learning Control
Distance neg_close zero close far
Angle medium
pos_big
pos_small
zero
neg_small
neg_big
Learning Method
Competitive Learning
Error Backpropagation
Variable Clustering
Learing Progress
0.1
0
3D-Picture
Distance
neg_close
zero
close
medium
far
Ok
IF
Angle
pos_big
pos_small
zero
neg_small
neg_big
Distance
neg_close
zero
close
medium
far
THEN
Power
neg_high
neg_medium
zero
pos_medium
pos_high
Aggregation...
Composition...
3-D Picture...
Spreadsheet...